IERG4210 Web Programming and Security (2022 Spring)
Assignment Marking Guidelines

Revision history

v1.0 Published on 28/03/2022

General Guidelines

The assignment is designed to let students practice what they have learned in the course. Students must be aware of web application security throughout web development. The whole assignment is split into 6 phases, leading all the way to a creative and functional shopping cart upon completion. Students should take a real-world website, parknshop.com, as a reference. In the assignment, students are expected to understand and apply proper security design principles and programming skills, regardless of which programming languages and libraries the students desire to use. The marking checklist included in the next page therefore outlines only the general requirements with a result-oriented basis in order to encourage students' creativity. For detailed guidance, students should refer to both lecture and tutorial notes.

Submission Policy

Students are required to package all of their source code, a README file, and any external resources (e.g. database, images, css and js files) into a zip file and submit it to the Blackboard. Each phase is associated with a firm submission deadline.

- Late Submission Penalty -- Late submission will lead to your mark reduction by the formula 0.9^n, where n is the round-up number of days delayed (e.g., assume your score is S and your submission is 9 hrs late → $0.9 \times S$, 25 hrs late → $0.81 \times S$, 49 hrs late →$0.729 \times S$, and so forth).
- *Final Demonstration* – Students will sign up for a timeslot to demonstrate their websites to a marker, who will then grade it according to the checklist. The marker will then evaluate the student's understanding with questions.

Honesty in Academic Work

CUHK places very high importance on honesty in academic work submitted by students, and adopts a policy of *zero tolerance* on cheating in examinations and plagiarism. Students are NOT allowed to submit anything that is plagiarised. Therefore, we treat every assignment our students submit as original except for source material explicitly acknowledged. We trust that students acknowledge and are aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website http://www.cuhk.edu.hk/policy/academichonesty.

IERG4210 Web Programming and Security (2022 Spring)
Assignment Marking Checklist v1

Phase 5: Secure Checkout Flow (Deadline: Apr 11 2022) (Subtotal: 20')

This is a tough phase, yet the most critical one, to escalate your website (building skill) to the next (professional) level. (You'll likely be offered a job if you can demonstrate such a level of web programming skills.) The implementation has already been outlined as below. Be prepared to spend a substantial amount of time debugging.

1. Sign up at https://developer.paypal.com and create two test accounts: _____ / 1'
    - A merchant account - after logging in to the Sandbox Test Site, modify necessary settings in the Selling Preferences under Profile
    - A buyer account – use it to pay for purchased items in your shopping portal
2. Enclose your shopping cart with a <form> element _____ / 3'
    - Use the Cart Upload Command of PayPal Website Payment Standard (cmd=_cart&upload=1)
    - Insert additional hidden fields that are required by PayPal (Read the first reference)
        - business, charset, currency_code, item_name_X, item_number_X, quantityX
        - invoice and custom
    - Create a checkout button that submits the form
3. When the checkout button is clicked: _____ / 4'
    - Pass ONLY the *pid* and *quantity* of every individual product to your server using AJAX and cancel the default form submission
    - Server generates a digest that is composed of at least:
        - Currency
        - Merchant's email address
        - A random salt
        - The *pid* and *quantity* of each selected product (Is quantity positive number?)
        - The current price of each selected product gathered from DB
        - **The total price of all selected products**

    Hint: separate them with a delimiter before passing to a hash function
    - Server stores all the items to generate the digest into a new database table called *orders*
        - The user could be logged in or as "guest" to purchase, store username with order in DB
    - Pass the lastInsertId() and the generated digest back to the client by putting them into the hidden fields of invoice and custom, respectively
    - Clear the shopping cart at the client-side
    - Submit the form now to PayPal using programmatic form submission
4. Setup an Instant Payment Notification (IPN) page to get notified once payment is completed
    - Validate the authenticity of data by verifying that it is indeed sent from PayPal _____ / 1'
        - Your IPN receiver page is served over HTTPS (using the SSL cert)
        - When contacting PayPal for the message authenticity check, use SSL and port 443
        - The sample code of validation protocol will be given in Tutorial 9

    Hint: sample code will be given in the tutorial
    - Check that txn_id has not been previously processed and txn_type is cart _____ / 1'
    - Regenerate a digest with the data provided by PayPal (same order and algorithm) _____ / 2'
    - Validate the digest against the one stored in the database table *orders* _____ / 2'
        - If validated, the integrity of the hashed fields is assured
        - Save the txn_id and ~~product list~~ (pid, quantity, and price) into DB

Debugging Hint: use error_log(print_r($_POST,true)) to print out the parameters passed by PayPal

5. After the buyer has finished paying with PayPal, auto-redirect the buyer back to your shop _____ / 1'
6. Display the DB *orders* table in the admin panel: product list, payment status…etc. _____ / 1'
7. Let members check what they have purchased in the most recent five orders. _____ / 4'

- Show the order information in the member portal.

References:
https://developer.paypal.com/docs/api-basics/sandbox
https://developer.paypal.com/docs/api-basics/notifications/ipn
http://www.evoluted.net/thinktank/web-development/paypal-php-integration